Squares: Supporting Interactive Performance Analysis for Multiclass Classifiers

Donghao Ren¹,², Saleema Amershi², Bongshin Lee², Jina Suh² and Jason D. Williams²

¹ University of California, Santa Barbara
² Microsoft Research, Redmond
Performance analysis is critical in machine learning

Data Collection → Feature Creation → Model Building → Performance Analysis
Performance analysis is critical in machine learning.
Performance analysis is critical in machine learning
Performance analysis is critical in machine learning
Common ways of performance analysis

- Summary statistics
  - Accuracy
  - Precision
  - Recall
  - Log-Loss
  - ...

- Confusion Matrix

```
Actual Class

Predicted Class

<table>
<thead>
<tr>
<th></th>
<th>89.3%</th>
<th>0.6%</th>
<th>6.9%</th>
<th>3.3%</th>
</tr>
</thead>
<tbody>
<tr>
<td>31.4%</td>
<td></td>
<td></td>
<td>19.4%</td>
<td>7.2%</td>
</tr>
<tr>
<td>18.6%</td>
<td>42.0%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.3%</td>
<td>1.1%</td>
<td>2.4%</td>
<td>80.2%</td>
<td></td>
</tr>
</tbody>
</table>
```

Problems

• Disconnected from the underlying data.
• Hide important information such as score distribution.
• Not trivial to support \textit{multiclass} classifiers.
Squares
Design Process

- Survey of Machine Learning Practices
- Design of Squares
- Controlled Experiment

Revise Design
Design Process

Survey of Machine Learning Practices → Design of Squares → Controlled Experiment

Revise Design
Design Process

Survey of Machine Learning Practices → Design of Squares → Controlled Experiment

Revise Design
Design Process

Survey of Machine Learning Practices → Design of Squares → Controlled Experiment

Revise Design
Design Goals

• **G1**: Show performance at multiple levels of detail to help practitioners prioritize efforts.
  - Overall / Class-level / Instance-level
  - Error severity (errors with higher score on the wrong class are more severe)

• **G2**: Be agnostic to common performance metrics.
  - Support a wider range of scenarios.

• **G3**: Connect performance to data.
  - Provide access to data. Use small visual footprint to reserve space for scenario-dependent data access views.
1. Each **class** is shown as a column

Dataset: **Glasses** from the UCI Machine Learning Repository
Visualization Design

1. Each **class** is shown as a column

2. Each **instance** is shown as a box

Dataset: **Glasses** from the UCI Machine Learning Repository
Visualization Design

1. Each class is shown as a column
2. Each instance is shown as a box
3. Instances are binned according to prediction scores

Dataset: Glasses from the UCI Machine Learning Repository
Visualization Design

Dataset: Glasses from the UCI Machine Learning Repository
Visualizing Count-Based Metrics: Overall Accuracy

- Accuracy: \( \frac{\text{Correct Predictions}}{\text{Total \# of Instances}} = \frac{\text{Correct}}{\text{Correct} + \text{Incorrect}} \)
Visualizing Count-Based Metrics: Class-Level

- Class-level precision and recall:

\[
\text{Precision: } \frac{TP}{TP + FP} = \frac{\square}{\square + \checkmark} \quad \text{Recall: } \frac{TP}{TP + FN} = \frac{\square}{\square + \square}
\]

FPs and FNs are comparably salient:
One-to-one correspondence between outlined boxes and striped boxes
Visualizing Score-Based Metrics

Higher scoring instance (more confident)

Lower scoring instance (less confident)

Worse score distribution
Help Prioritizing Debugging Efforts

More severe error (confidently wrong)

Less severe error (prediction can flip if scores change slightly)
Visualizing Confusion Between Classes

C5 is confused with C3
Instance-Level Details

On-hover parallel coordinates for detailed scores

Dataset: MNIST Handwritten Digits
Scalability

Each strip represents 10 boxes

Boxes

Strips

Stacks

Truncation indicators
Scalability

Toggle between 3-levels of aggregation
Evaluation
Controlled Experiment

• 24 participants

• Part 1: Comparison
  • Compare Squares against a commonly used Confusion Matrix
  • Within-subject design

• Part 2: (Squares Only) Score Distribution
  • Evaluate Squares’ ability to convey score distribution
Part 1: Squares vs. Confusion Matrix

Squares with a Sortable Table

Confusion Matrix with a Sortable Table

Select/Deselect individual cells.
Select cells of a given row/column.
Part 1: Tasks

• T1 – Overall
  • Select the classifier with the larger number of errors

• T2 – Class-level
  • Select one of the two classes with the most errors

• T3 – Instance-level
  • Select an error with a score of .9 or above in the wrong class
Part 1: Squares Performed Better

• Task Time

Squares lead to faster task time  
(Main Effect: p < 0.001)  

Squares scale better in terms of the  
number of classes  
(Interaction Effect: p = 0.012)
Part 1: Squares Performed Better

- **Accuracy**

  - Squares lead to more accurate results

  \[(p < 0.001)\]
Part 1: People Preferred Squares

Squares was more helpful

Squares was preferred
Part 2: (Squares Only) Distribution Tasks

• T4 – Overall
  • Select the classifier with the worst distribution

• T5 – Class-level
  • Select one of the two classes with the worst distribution

• T6 – Confusion
  • Select the two classes most confused with each other
Part 2: Squares was helpful in distribution tasks

Task Time (s)

Accuracy

Helpfulness
Freeform Feedback

• Positive:
  • “Granular and at the same time general overview of the classifiers is great.”
  • “Seeing the distribution of scores is very helpful.”
  • “Had fun for the first time while classifying!”

• Negative:
  • “I prefer having numbers than pure display.”
  • “[Confusion Matrix is] more straightforward, lower learning curve.”
Future Work

• Further Evaluation
  • Compare to alternative designs of Confusion Matrix, as well as other visualization designs in the literature

• Scalability
  • Supporting more than 20 classes
  • Optimizing color assignments
Squares as a Tool

- Deployed along with a machine learning toolkit within Microsoft
Acknowledgements

• We thank the support and feedback from the Machine Teaching Group at Microsoft Research.

• We thank the anonymous reviewers for their constructive comments.
Thanks! Questions?

Donghao Ren (donghao.ren@gmail.com)
University of California, Santa Barbara
Survey of Machine Learning Practices

- Survey within a large software company in July. 2015.
- 102 respondents:

Respondents’ Roles in the company

- Data scientist: 40%
- Software engineer: 30%
- Researcher: 10%
- Program manager: 10%
- Other: 10%
Number of Classes

• How many classes do your classifiers typically deal with (check all that apply)?
  • Most respondents typically deal with less than 20 classes.
Important Tasks

“How difficult” and “how important” ratings of tasks:

- Prioritizing efforts is difficult even for expert users.
- Understanding instance-level performance is relatively more difficult in common tools.
Integrating into LUIS (Language Understanding Intelligent Service)